**Statement Testing**

**History.**

Although Oracle provides for AR statements, it does not meet the needs of LifeWay. Around 2004-2005, while LifeWay was still using Vista, major changes were made to the look and feel of Lifeway statements. During the Oracle implementation that went live in 2006, a major effort was made to keep that statement functionality. The only thing we use from the Oracles standard statement functionality is the ar.ar\_statement\_cycles table, which is accessed from the Receivables All responsibility. We have a record there for each of our statement waves: W1, W2, W3, W4, W5, WC, and WF. Each statement wave is generated once a month. Each customer is assigned to a particular statement wave. That assignment can be located by the following query:

select wav.name

from ar.hz\_cust\_accounts hca,

ar.hz\_customer\_profiles hcp,

ar.ar\_statement\_cycles wav

where hca.account\_number = '2000049678' -- for example

and hca.cust\_account\_id = hcp.cust\_account\_id

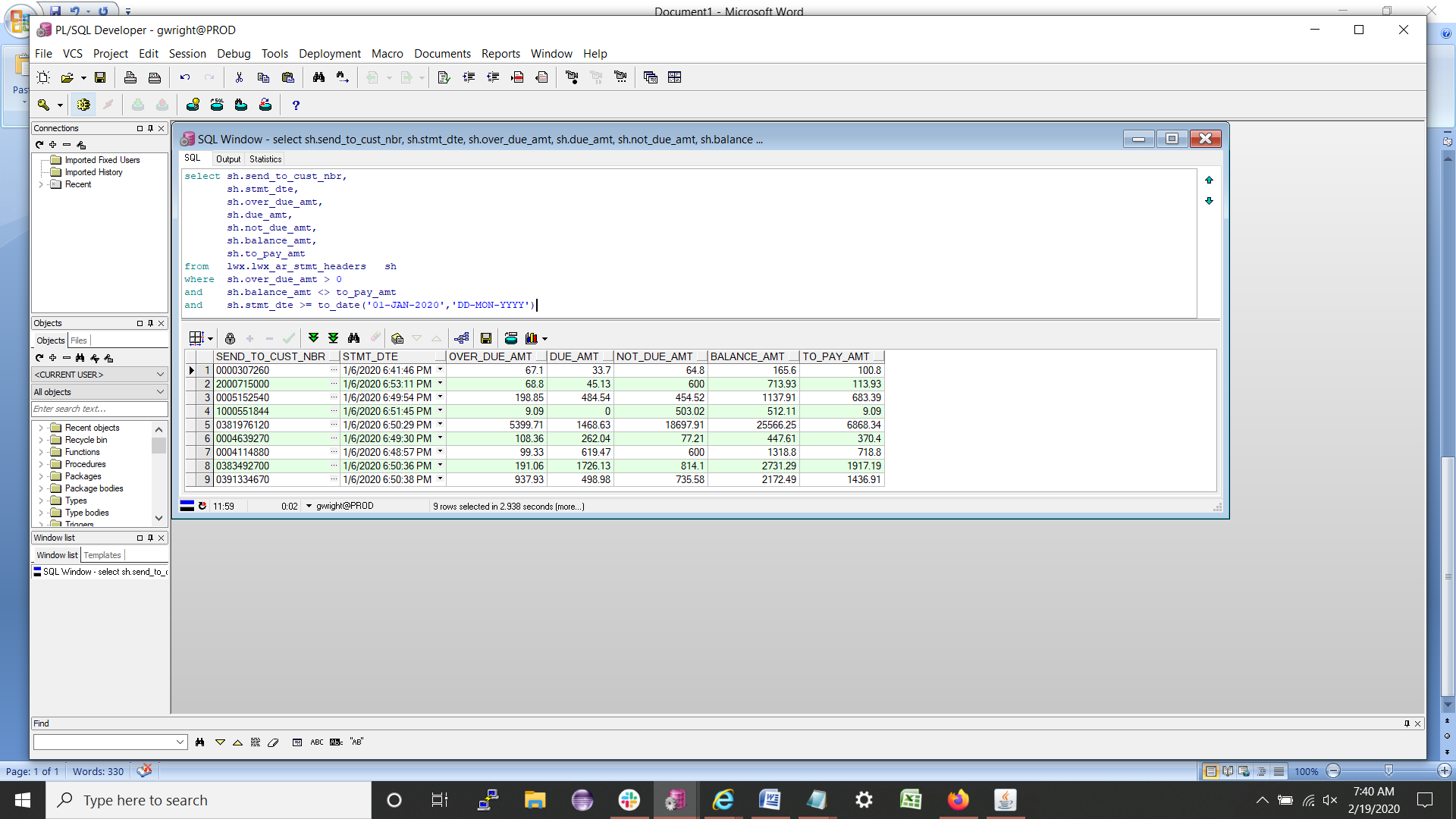
and hcp.site\_use\_id is null

and hcp.statement\_cycle\_id = wav.statement\_cycle\_id

**Archiving:**

Three customer tables are used to retain statement data: lwx.lwx\_ar\_stmt\_headers, lwx.lwx\_ar\_stmt\_lines, and lwx.lwx\_ar\_stmt\_line\_details. The printed statement has four sections: the summary page (which appears first), the pre-paid page (which only appears if needed), the detail page (one line for each invoice, debit memo, credit memo, or payment), and the consolidated invoice section (one or more pages for each invoice). The things printed on the detail page correspond to lwx.lwx\_ar\_stmt\_headers. There is one of these for each customer’s statement. For your project, here are the most important columns on that record:

* STMT\_HDR\_ID – Key that uniquely identifies the statement and connects lwx.lwx\_ar\_stmt\_headers to lwx.lwx\_ar\_stmt\_lines.
* STATEMENT\_CYCLE\_ID – Key that links to a key by the same name in ar.ar\_statement\_cycles.
* STMT\_RUN\_CONC\_REQ\_ID – This column records the request\_id from the LWX AR Consolidated Statement Generation Program that produced this set of statements.
* STMT\_DTE – This column records the date the statement was created.
* OVER\_DUE\_AMT – This is how much was due from the last statement and remains unpaid.
* DUE\_AMT – This is how much is due now that did not appear on the last statement.
* TO\_PAY\_AMT – This will be the sum of over\_due\_amt and due\_amt.
* STMT\_DUE\_DTE – This date is set 25 days out from stmt\_dte.
* NOT\_DUE\_AMT – This amt is from invoices that will not be due until some time in the future.
* BALANCE\_AMT – This is the sum of everything on the customer’s account, whether it is due or not.
* SEND\_TO\_CUST\_NBR



The above screen shows how the various amounts go together.

Using stmt\_hdr\_id the lwx.lwx\_ar\_stmt\_headers table is joined to lwx.lwx\_ar\_stmt\_lines. The table lines table has three kinds of lines, depending on the value of the rec\_type\_cde column. If rec\_type\_cde = ‘F2’ then the line is part of the prepaid section of the statement. If the rec\_type\_cde = ‘F3’ then the line is part of the detail section of the statement. If the rec\_type\_cde = ‘F4’ then there is an invoice consolidated with the statement. To find statements that actually have consolidated invoices you can run a query like this:

select sh.send\_to\_cust\_nbr,

sh.stmt\_hdr\_id,

sh.stmt\_dte,

sh.invo\_page\_cnt,

sh.total\_page\_cnt

from lwx.lwx\_ar\_stmt\_headers sh

where sh.stmt\_dte >= to\_date('01-JAN-2020','DD-MON-YYYY')

and sh.invo\_page\_cnt > 0

Then, to see how those lines are logically connected between all three tables you could run this query:

select sl.trans\_nbr,sd.\*

from lwx.lwx\_ar\_stmt\_headers sh,

lwx.lwx\_ar\_stmt\_lines sl,

lwx.lwx\_ar\_stmt\_line\_details sd

where sh.stmt\_hdr\_id = 8506376 – for example

and sh.stmt\_hdr\_id = sl.stmt\_hdr\_id

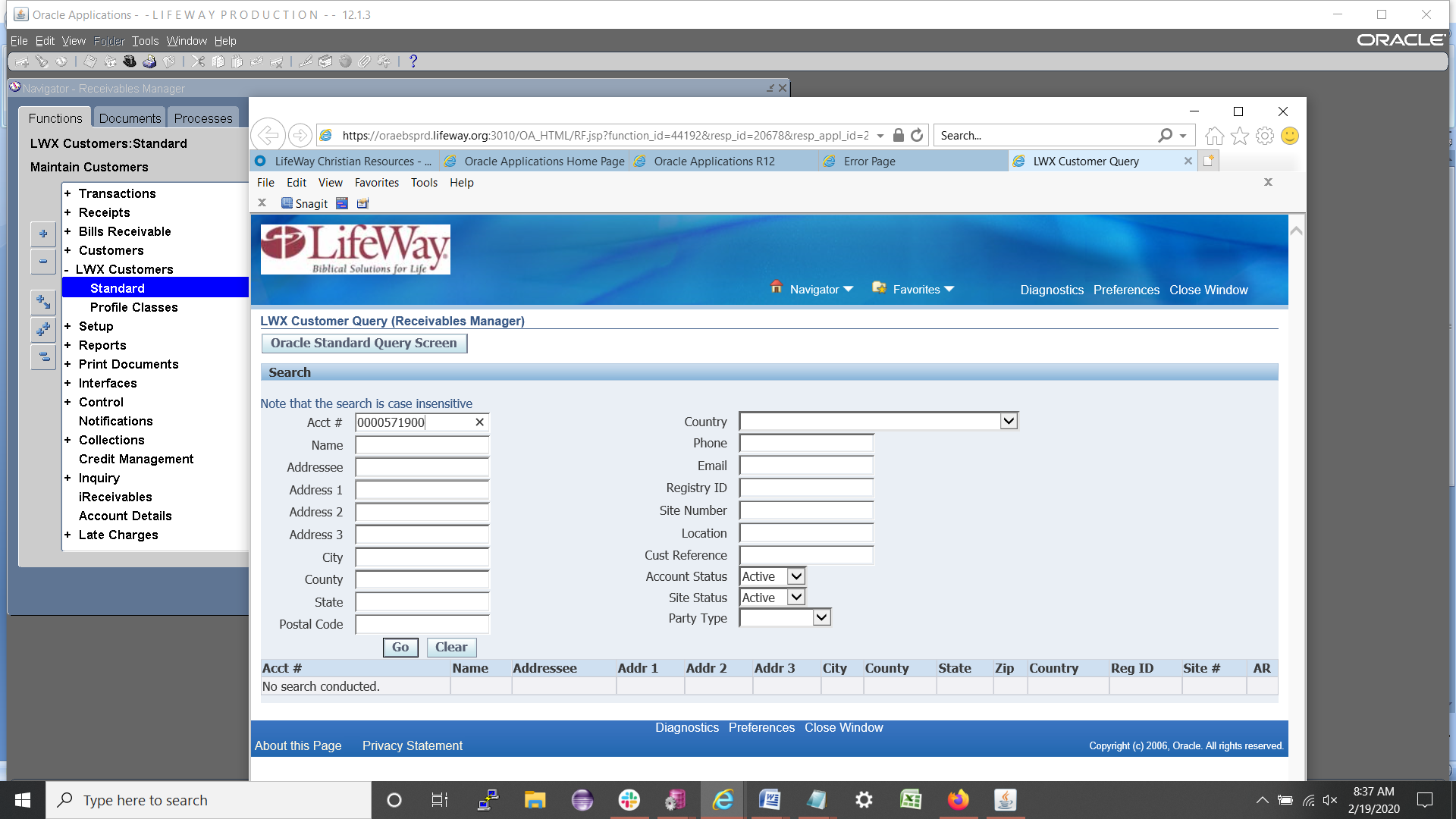
and sl.rec\_type\_cde = 'F4'

and sl.stmt\_line\_id = sd.stmt\_line\_id

So it is that these three tables record all we need to archive and reproduce a statement.

**Reproducing a statement.**

The screen snippet below shows the initial navigation. From responsibility Receivables All or LWX AR Manager select LWX Customers, select Standard, and the form on the right will come up. Fill in the account number and click go.



Then from the top click on LWX AR Acct Summary. A new form will come up.

Under Statement History select the statement you want. It will open up as a PDF file. While in the DEV environment you will be able to modify stmt\_dte to get the statement testing scenarios you need.

**How statements are produced.**

During the AR Dayend the request set LWX AR Auto Invoicing Set runs LWX AR WRAPPER STMT. That program read the ar.ar\_statement\_cycles table to see if there is a statement wave that matches today’s date. If there is it runs LWX AR Consolidated Statement Generation Program. When that program finishes the lwx\_ar\_stmt\_headers and associated tables are done. After that the wrapper runs other programs and scripts that have more to do with formatting the statements for DNI, which is our printer and mailer. Meanwhile, you can run LWX AR Consolidated Statement Generation Program directly, for a single account number. Before you run a customer you will want to set the statement cycle for that customer to today’s date. Here is how:

From Receivables All select SETUP > PRINT > STATEMENT CYCLES > Ctrl F11

Find the statement cycle for that customer. Do this by finding the current month and year in the list. Change the day to today’s date.

Alert: Make sure the customer owes at least $5 or the statement won’t print. Also, make sure no other statement wave is set to today’s date. If it is, change the date in that other wave. Also, make sure a statement has not been produced for that customer in the last 25 days, or it won’t print. Manipulate lwx\_ar\_stmt\_headers.stmt\_dte on the last statement if you need to.

If you need to get rid of a statement you created you can go to the LWX-Apps Developer and run LWX AR Delete Statements. You have two ways to delete a statement, but you only need one of them.

This should be enough to get you started thinking about how to test.